#22

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re U.S. Patent Application of)
)
Isom)
)
Serial No.: 08/905,701) Examiner: George L. Opie
Filed: August 4, 1997) Group Art No.: 2151
)
For: **Method of Sequencing Computer Controlled Tasks Based on the Relative Spatial Location of Task Objects in a Directional Field**)
)
Attorney Docket No. 4210-001)

Raleigh, North Carolina
November 2, 2001

Commissioner For Patents
Washington, D.C. 20231

Sir:

The present appeal brief is filed in triplicate pursuant to 37 C.F.R. § 1.192. Applicant also encloses a check in the amount of $360 .00 to cover the $160.00 appeal brief fee required under 37 C.F.R. § 1.17(c), and the $200.00 fee required for a two-month extension of time required under 37 C.F.R. § 1.17(a)(2). Applicant is a small entity. If additional fees are required, the Commissioner is authorized to charge Deposit Account 18-1167.

APPEAL BRIEF

(1) REAL PARTY IN INTEREST

The real party in interest is Applicant, Steve Isom.

(2) RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences to the best of Applicant's knowledge.

(3) STATUS OF CLAIMS

Forty-one (41) claims have been presented for examination. Applicant originally filed claims 1-28, cancelled claims 26-28 without prejudice, and added claims 29-41 by amendment. Claims 1-25 and 29-41 are pending, and Applicant appeals from the rejection of these claims.

(4) STATUS OF AMENDMENTS

All amendments have been entered.

(5) SUMMARY OF INVENTION

Applicant's invention provides a graphical method of sequencing computer controlled tasks based on the relative positioning of task objects in two or more spatial dimensions within a directional field. Task objects appear as icons on the user's computer screen, with each task object being associated with one or more desired tasks. A task might be one or a particular group of program instructions, or one or more control events. In general, Applicant's invention as claimed provides novel sequencing control in a graphical task-sequencing environment.

Sequencing of tasks within the framework of Applicant's invention as claimed depends upon the relative positions of task objects arranged within the directional field. The directional field is at least two-dimensional and may be presented to the user on a computer display as a window or workspace in which the task objects may be arranged. In some embodiments, the directional field takes on or at least simulates attributes of higher dimensionality, allowing users

to arrange task objects in more than two dimensions. In any case, the directional field has an associated directional attribute that determines the particular task sequencing applied to a given arrangement of task objects within the directional field. For example, a given directional attribute setting may cause tasks to be sequenced from left to right and from top to bottom within the directional field. Of course, the user is free to modify the directional attribute such that tasks are sequenced in any desired direction.

This ability to set or change the directional attribute at will imparts a wholly novel approach to sequencing and re-sequencing of computer controlled tasks. Within the framework of Applicant's invention, a given arrangement of task objects within the directional field represents not a single task sequence as in the prior art approaches, but rather represents a potentially large set of distinct task sequences selectable by simply setting or modifying the directional attribute. With this approach, the user can explore many different task sequencing relationships without rearranging task objects in the directional field. For example, the user may change the directional attribute from a left-right sequencing sense to a right-left sequencing sense independent of or in addition to changing the top-bottom sequencing sense. One may readily see that as the directional field takes on higher dimensionality, the sequencing permutations associated with a single arrangement of task objects within the directional field increases exponentially.

(6) ISSUES

The issue is whether claims 1-25 and claims 29-41 are obvious under 35 U.S.C. § 103(a) in light of Carlson et al., Keller et al., and Ingalls et al.

(7) GROUPING OF CLAIMS

Claims 1-25 stand or fall together, and claims 29-41 stand or fall together.

(8) ARGUMENT

A. Introduction

The prior art of record in any combination does not teach or suggest the subject matter of the present invention as claimed, and thus does not render the present invention obvious within the meaning of 35 U.S.C. § 103(a).

B. Summary of Prior Art

U.S. Patent No. 5,623,592 to Carlson et al. (Carlson) is directed to controlling external devices using a computer. Carlson provides an experiment design region on the user's computer screen which appears, for example, as a window to the computer user. Carlson allows the user to graphically draw or place a timeline within the window. The timeline has a fixed directional sense. That is, a horizontal timeline may represent a left-to-right time flow or a right-to-left time flow. Similarly, a vertical timeline might represent a top-to-bottom time flow or a bottom-to-top time flow. In any case, once the user places a timeline in the window, its directional sense is fixed. The user then arranges task icons of different types along the timeline such that the task icons are arranged according to a desired task sequencing. Because the directionality of the Carlson timeline is fixed, the only way to change task sequencing within a particular set of task icons is to physically rearrange the icons arrayed along the timeline.

The examiner's arguments make too much of the "general sequencing rule" discussed in Carlson. Carlson defines a number of different types of task icons, one type being sequence control icons comprising kinetic icons, stop icons and sample icons. Col. 14, lines 41-46. While

Carlson does state that the "sequence of operations is determined according to a general sequencing rule, as modified by any sequence control icons in the icon sequence," a careful reading of Carlson reveals that this remark simply identifies that sequence control icons can stop a sequence, can cause a sequence to branch into another sequence or can cause a portion of a particular sequence to repeat a desired number of times.

However, in no case, is the time sense of the timeline modified, nor is the sequence direction ever changed. For example, so-called "kinetic" icons simply specify the number of time selected portions of the task sequence are repeated, but do nothing to alter the underlying sequential direction of task execution. Col. 14, line 54 – Col. 15, line 23. Under all circumstances in Carlson, if the user desires to change the task execution sequence associated with a particular arrangement of icons on the timeline, the user absolutely has to change the arrangement of icons along the timeline.

U.S. Patent No. 5,767,852 to Keller et al. (Keller) is directed to prioritizing program or process execution within multitasking computer operating systems, such as UNIX or IBM's AIX computer operating systems. User's place process icons within a graphical window in a desired arrangement relative to a priority controller icon. The positioning of process icons relative to the process controller icon determines the priority assigned to the associated process running within the multitasking environment. Keller is not directed to sequencing tasks. Rather, Keller is directed to managing the cooperative execution of a number of programs or processes in a multitasking environment.

Keller discloses that process prioritization may be based on arrangements made within a graphical workspace of two or more dimensions. Thus, the process controller icon may simply be a two dimensional bar, or may be, for example, a simulated three-dimensional workspace.

The point of Keller is to provide a graphical mechanism for assigning process priorities within a multitasking operating system, and not to provide task sequencing control for a group of computer controlled tasks.

"Fabrik" is the name given to a visual programming environment developed by Dan Ingalls, et al. (Ingalls). Fabrik represents computer instructions as boxes in a graphical user interface, which the user "wires" together to build more complex functions, or entire computer programs that are composites of the computational elements represented by the wired-together boxes. OOPSLA '88 Proceedings (Proceedings), p. 176, ¶ 3. A typical computational element within the Fabrik library might comprise a string viewer or a standard mathematical computation. Defining a program instruction flow (a task sequence) within Fabrik requires the user to manually connect one box to another. Fabrik refers to these connections as "wires," and the user implements a desired task sequence by wiring a collection of boxes together in a desired order. The spatial arrangement of boxes within the Fabrik graphical workspace has nothing to do with the task sequencing associated with those boxes. Task sequencing is entirely driven by the manual wired connections drawn by the user.

Fabrik defines connection wires as unidirectional or bi-directional. That is, a connection drawn by the user between two boxes within the workspace might represent a one-way task sequence or a two-way task sequence, depending on the user's selection of wire type, and on the underlying definition of program instructions within the interconnected boxes. The definition of program instructions within the interconnected boxes is important because for Fabrik to implement its so-called bi-directional connection, each of the bi-directionally interconnected boxes must actually have two distinct sets of program instructions. The first set of instructions in each box is associated with one direction of task flow, while the second set of instructions in

each box corresponds to the opposite direction of task flow. The practical upshot of this implementation is that Fabrik uses a bi-directional graphical wire to represent two separate task flows. Figures 2a and 2b provide clear illustration of this implementation detail.

C. The Instant Invention Is Non-Obvious Over The Prior Art

The examiner sites several references (i.e., Carlson, Keller, and Ingalls), none of which disclose changing the ordered computer-controlled task sequences by simply changing a directional attribute of a spatial field. None of the sited references, whether taken individually or in any combination, disclose or suggest the claimed invention. Applicant's invention as claimed discloses an environment wherein users can change the task sequencing associated with a graphical arrangement of task objects based simply on changing or adjusting a directional attribute. Thus, one spatial arrangement corresponds to multiple task sequences.

Carlson allows users to arrange task icons on a fixed timeline within a scheduling window. The positioning of task icons along the timeline entirely determines the execution sequence of the tasks represented by those icons. If the user desires a different execution sequence, the user is forced to rearrange the task icons along the timeline in accordance with the desired sequence change. The directional rule defining Carlson governs the time sense or task execution direction of any timeline placed within the window, and once placed, the timeline has a fixed execution direction. Nothing in Carlson discloses, suggests, or even alludes to a scenario in which a given arrangement of task icons along a timeline might be subjected to a reversal or other change in execution sequence based on changing the time sense of the timeline.

In Keller, the examiner finds the idea of using spatial relationships between process icons arranged in two or more dimensions to prioritize program, or process execution in a multitasking environment. Keller is not directed to task sequencing, and nothing in Keller discloses the idea

of applying a changeable task sequence to a graphical arrangement of task icons. In short, the examiner looks to Keller for the idea of working in two or more dimensions but this is not the point of the Applicant's invention.

The examiner then looks to Ingalls for the bare idea of bi-directionality. Bi-directionality is not the point of the Applicant's invention. Within the framework of Applicant's claimed invention, a given arrangement of task icons represents a plurality of possible task sequences that the user is free to select and experiment with based simply upon modifying an associated directional attribute without need for rearranging any of the task icons. The Fabrik programming environment disclosed by Ingalls has nothing to do with task sequencing based on the arrangement of task icons, and in fact requires the user to define a desired task (program) sequence by explicitly drawing wire connections between task icons (boxes). Thus, the underlying method of defining a task sequence in Ingalls is antithetical to the approach used in the instant invention. One point of the Applicant's invention as claimed is to obviate the need for the user to draw explicit task sequencing links as taught by Fabrik.

Beyond contrasting concepts embodied by the Applicant's claimed invention and the programming environment of Fabrik, the bi-directional link in Fabrik does not even represent a bi-directional task sequence. While this is a minor point, it is further evidence that the examiner's use of Fabrik to support an allegation of obviousness regarding the claimed invention is flawed. In short, Carlson teaches a timeline that fixes the task execution sequence for a group of task icons arranged along that timeline, and does not even suggest a technique whereby the same spatial arrangement of task icons might yield different task sequences. Keller does not relate to task sequencing, but rather relates to process prioritization within a multitasking operating system. Keller associates process priorities with a given arrangement of process icons

within a workspace. Keller does not disclose or even intimate associating different prioritization schemes with a given arrangement of process icons. Lastly, Ingalls discloses a visual programming environment that requires the user to draw manual connections between task icons in a graphical window. The order of execution associated with the icons is fixed explicitly on the manually drawn wire connections between icons.

Conclusion

The references sited by the examiner do not in any combination disclose or suggest the present invention. As the references do not teach or even suggest the novel aspects of Applicant's invention as claimed, the cited references cannot, under any circumstance, support an obviousness rejection. Applicant has carefully drawn the pending claims to incorporate the novel aspect of allowing a user to apply different task execution sequences to a given spatial arrangement of task icons within a directional field. Despite the examiner's assertions regarding the cited references, none of the cited references in any combination disclose this novel approach to task sequencing.

Respectfully submitted,

COATS & BENNETT, P.L.L.C.

By: Michael D Murphy

Michael D. Murphy
Registration No. 44,958

P.O. Box 5
Raleigh, NC 27602
Telephone: (919) 854-1844

CERTIFICATE OF MAILING
I HEREBY CERTIFY THAT THIS DOCUMENT IS BEING DEPOSITED WITH THE UNITED STATES POSTAL SERVICE AS FIRST CLASS MAIL, POSTAGE PREPAID, IN AN ENVELOPE ADDRESSED TO: COMMISSIONER FOR PATENTS, WASHINGTON, D.C. 20231.

SIGNATURE Michael D Murphy

DATE 11/02/01

(9) APPENDIX

Claims

1. A method for sequencing a plurality of tasks performed or controlled by a computer, comprising:

a) placing task objects in an at least two-dimensional directional field having a user-changeable directional attribute, wherein said task objects represent the tasks to be performed by said computer and;

b) sequencing, by said computer, of one or more of the task objects in the directional field based on the relative spatial location of the task objects in the directional field and the directional attribute of the directional field..

2. The sequencing method of claim 1 further including the step of re-sequencing said task objects by changing the relative spatial location of the task objects in the directional field.

3. The sequencing method of claim 1 further including the step of selecting a directional attribute for the directional field.

4. The sequencing method of claim 1 wherein said task objects have one or more modifiable properties for controlling the behavior of the task objects.

5. The sequencing method of claim 4 wherein one of said modifiable properties is used to include or exclude a task object in the directional field from said sequence.

6. The sequencing method of claim 4 wherein at least one of the modifiable properties specifies the tasks to be performed by the task object.

7. The sequencing method of claim 1 further including the step of placing, a master object in the directional field for initiating said sequence of tasks.

8. The sequencing method of claim 7 wherein said task objects are responsive to said master object to perform their associated tasks.

9. The sequencing method of claim 8 further including the step of defining a limited region of influence for said master object, wherein said master object is used to initiate a sequence including task objects falling within the region of influence of the master object.

10. The sequencing method of claim 9 wherein the sequence includes only those task objects falling within the region of influence of the master object.

11. The sequencing method of claim 8 including a plurality of master objects, each of which has a region of influence, wherein at least one master object is responsive to the activation of one or more other master objects to initiate sequencing of task objects within its own control region.

12. The sequencing method of claim 8 further including the steps of selecting a type for each task object from a list of pre-defined types, wherein each master object is programmed to sequence only task objects of certain specified types.

13. The sequencing method of claim 12 further including the step of defining a general type for master objects for sequencing task objects of all types.

14. A method for sequencing a plurality of tasks performed or controlled by a computer comprising:

a) displaying on a computer display a user interface having an at least two-dimensional directional field;

b) placing in response to user input, task objects in said directional field, wherein said task objects represent the tasks to be performed by said computer;

c) selecting, by a user, a directional attribute for said directional field;

d) sequencing, by said computer, of one or more task objects in the directional field based on the relative spatial location of the task objects in the directional field and the directional attribute of the directional field.

15. The sequencing method of claim 14 further including the step of re-sequencing said task objects by changing the relative spatial location of the task objects in the directional field.

16. The sequencing method of claim 14 wherein said task objects have one or more modifiable properties for controlling the behavior of the task objects.

17. The sequencing of claim 16 wherein one of said modifiable properties is used to include or exclude a task object in the directional field from said sequence.

18. The sequencing method of claim 16 wherein at least one of the user-definable properties is used to specify the tasks to be performed by the task object.

19. The sequencing method of claim 14 further including the step of placing, a master object in the directional field for initiating said sequence of tasks.

20. The sequencing method of claim 19 wherein said task objects are responsive to said master object to perform their associated tasks.

21. The sequencing method of claim 20 further including the step of defining a limited region of influence for said master object, wherein said master object is used to initiate a sequence including task objects falling within the region of influence of the master object.

22. The sequencing method of claims 21 wherein the sequence includes only those task objects falling within the region of influence of the master object.

23. The sequencing method of claim 20 including a plurality of master objects, each of which has a region of influence, wherein at least one master object is responsive to activation of one or more other master objects to initiate sequencing of task objects within its own region of influence.

24. The sequencing method of claim 20 further including the steps of selecting a type for each task object from a pre-defined list of types, wherein each master object is programmed to sequence only task objects of certain specified types.

25. The sequencing method of claim 24 further including the step of defining a general type for master objects for sequencing task objects of all types.

29. A method for sequencing a plurality of tasks performed or controlled by a computer, comprising:

a) placing task objects in a directional field having at least two dimensions and a user-changeable directional attribute, wherein said task objects represent the tasks to be performed by said computer; and

b) sequencing, by said computer, of one or more of the task objects in the directional field based on the relative spatial location of the task objects in the directional field and the directional attribute of the directional field.

30. The sequencing method of claim 29 further including the step of re-sequencing said task objects by changing the relative spatial location of the task objects in the directional field.

31. The sequencing method of claim 29 further including the step of selecting a directional attribute for the directional field by a user.

32. The sequencing method of claim 29 wherein said task objects have one or more modifiable properties for controlling the behavior of the task objects.

33. The sequencing method of claim 32 wherein one of said modifiable properties is used to include or exclude a task object in the directional field from said sequence.

34. The sequencing method of claim 32 wherein at least one of the modifiable properties specifies the tasks to be performed by the task object.

35. The sequencing method of claim 29 further including the step of placing a master object in the directional field for initiating said sequence of tasks.

36. The sequencing method of claim 35 wherein said task objects are responsive to said master object to perform their associated tasks.

37. The sequencing method of claim 36 further including the step of defining a limited region of influence for said master object, wherein said master object is used to initiate a sequence including task objects falling within the region of influence of the master object.

38. The sequencing method of claim 37 wherein the sequence includes only those task objects falling within the region of influence of the master object.

39. The sequencing method of claim 36 including a plurality of master objects, each of which has a region of influence, wherein at least one master object is responsive to the activation of one or more other master objects to initiate sequencing of task objects within its own control region.

40. The sequencing method of claim 36 further including the steps of selecting a type for each task object from a list of pre-defined types, wherein each master object is programmed to sequence only task objects of certain specified types.

41. The sequencing method of claim 40 further including the step of defining a general type for master objects for sequencing task objects of all types.